Asynchronous Stereo Vision for Event-Driven Dynamic Stereo Sensor Using an Adaptive Cooperative Approach

Ewa Piątkowska, Ahmed Nabil Belbachir,
Member IEEE,
Safety and Security Department
AIT Austrian Institute of Technology
Donau-City Straße 1/5, A-1220 Vienna, Austria
{ewa.piatkowska.fl;nabil.belbachir}@ait.ac.at

Margrit Gelautz, *Member* IEEE
Institute for Software Technology
and Interactive Systems
Vienna University of Technology
Favoritenstraße 9-11, A-1040 Vienna, Austria
gelautz@ims.tuwien.ac.at

Abstract

This paper presents an adaptive cooperative approach towards the 3D reconstruction tailored for a bio-inspired depth camera: the stereo dynamic vision sensor (DVS). DVS consists of self-spiking pixels that asynchronously generate events upon relative light intensity changes. These sensors have the advantage to allow simultaneously high temporal resolution (better than 10μs) and wide dynamic range (>120dB) at sparse data representation, which is not possible with frame-based cameras. In order to exploit the potential of DVS and benefit from its features, depth calculation should take into account the spatiotemporal and asynchronous aspect of data provided by the sensor. This work deals with developing an appropriate approach for the asynchronous, event-driven stereo algorithm. We propose a modification of the cooperative network [6] in which the history of the recent activity in the scene is stored to serve as spatiotemporal context used in disparity calculation for each incoming event. The network constantly evolves in time - as events are generated. In our work, not only the spatiotemporal aspect of the data is preserved but also the matching is performed asynchronously. The results of the experiments prove that the proposed approach is well suited for DVS data and can be successfully used for our efficient passive depth camera.

1. Introduction

3D vision is currently intensively investigated in computer vision, thanks to the introduction of new depth cameras in consumer markets (e.g. Kinect). Different technologies are used for depth acquisition including passive and active methods. In the former depth is inferred from the parallax - displacements between two or more views of the same scene (e.g. stereo reconstruction, structure from motion). The latter is based on the analysis of the emitted light (laser, infrared or light patterns) with active triangulation methods or time-of-flight measurements. In this work we deal with the stereo dynamic vision sensor, therefore the challenge is to design an efficient stereo algorithm adapted for specific features of the sensor.

Dynamic vision sensors can play an important role in the next generation depth cameras and in some aspects can overcome the limitations of the other depth cameras. Firstly, they offer very high temporal resolution achieved by asynchronous data generation (benefits of frame-free vision). Secondly, their wide dynamic range and sensitivity to relative light intensity change allows for outdoor applications under uncontrolled lighting conditions. What is more, dynamic vision sensors are efficient; they capture only the prominent features of the scene (edges), reduce data redundancy (pixels spike only when a change is detected) and their low power consumption makes them applicable for mobile robots. Additionally, DVSs are designed for industrial applications and are suited for long-time operation.

Several attempts have been made to address the problem of 3D reconstruction for DVS. One typical approach [9][10] was to apply conventional stereo matching on pseudo-frames rendered from DVS events accumulated over a particular period of time. Nevertheless, the biological character of the data provided by DVS imposes different methods of processing than those proposed in the conventional computer vision. Moreover, transformations into the image-like representation may lessen the benefits of asynchronous vision. Therefore, it is more suitable to asynchronously process data in the same form as they are delivered by the sensor without losing the high temporal resolution of events occurrence. We claim that in order to exploit the full potential of dynamic vision sensors, the asynchronous aspect of the events should be preserved to better mimic the mechanism in biological vision. In the proposed algorithm we use the early model for cooperative stereo computation introduced by Marr and Poggio [6]. Furthermore the possibility to adapt the cooperative network for dynamic matching of events is investigated.

The paper is structured as follows: firstly we briefly show the data from dynamic vision sensors and present approaches proposed in the literature for the DVS stereo reconstruction. Then, we introduce our concept of

cooperative event-based stereo matching along with the description of the algorithm. The experimental proof of concept is given in Section 4. We conclude with a short summary.

2. Stereo for dynamic vision sensors

This section starts with a short description of the dynamic vision sensor properties and characteristics. Afterwards, related works on stereo vision using this type of sensors are listed in order to motivate the purpose of this research.

2.1. Short review of dynamic vision sensors

Dynamic vision sensors consist of self-spiking pixels which are sensitive to relative change in the light intensity. The sensor encodes visual information as a stream of events where each event represents a pixel's activity (spike) at a particular time. As depicted in Figure 1, there is significant difference in data captured by a conventional camera (Figure 1(a)) and dynamic vision sensors (Figure 1(b-c)). Moreover, dealing with a static sensor setup, only the scene dynamics is captured. Thus each moving object generates a cloud of events in space and time.

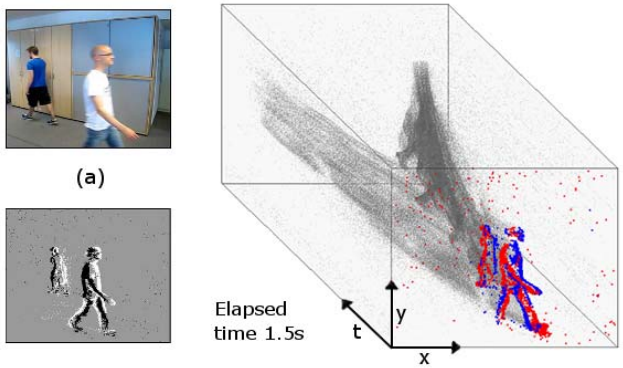

Figure 1: (a) Dynamics of two people walking in the room captured by a conventional video camera. The same scene captured by the dynamic vision sensor results in the stream of events shown in (b) image-like and (c) spatiotemporal representation (adapted from [1]).

The additional information attached to the event, despite of location and time, is the polarity which denotes the intensity increase (ON) or decrease (OFF). The data stream from DVS is encoded with Address Event Representation (AER). Each event *TAE* (Time stamped Address Event) is defined by a tuple:

$$TAE =< x, y, t, p, ch >, \quad (1)$$

where *(x,y)* is the pixel address, *t* is the event timestamp, *p* is the polarity (ON / OFF), and the channel (*ch*) denotes the source of the event (left or right view).

The stereo matching task is to find correspondences between the left and right events stream. The desired stereo algorithm output is to compute for each event in the stream an assigned disparity value d, resulting in the tuple:

$$TAE_{3D} =< x, y, d, t, p, ch > \quad (2)$$

2.2. Related work

The first attempt for event-based stereo vision was proposed by Mahowald and Delbruck [5]. Inspired by the cooperative algorithm [6] and its applicability for the on-chip implementation, they developed a stereo vision chip for 1D image matching. Although the results of this on-chip stereo were very promising, the main drawback of hardware approaches is the lack of flexibility, e.g. fixed disparity resolution (as defined by the size of the correlation network). Moreover, for 2D pixel arrays, the complexity of the chip's architecture would drastically increase.

The cooperative approach was also an inspiration for software solutions for the DVS stereo. Hess [3] proposed to use the cooperative matching for each row in the image. Additionally, he introduced a time-based matching function to assign some initial weights for each possible match. According to the observation that correct matches are more likely to be coincident in time, the weight of the match is higher for those events which are closer in time. The disparity is assigned by the highest value of high-weighted matches. Such algorithm, however, can only be reliable under the assumption of homogeneous disparity of the scene, which can be hard to obtain in real world scenarios. Nevertheless, it can be successfully used for camera alignment. In a further investigation, Hess [3] introduced spatiotemporal window matching for event-based stereo. In this approach, the correspondence between events is established on the basis of the similarity of the spatiotemporal neighborhood.

The importance of timing information in event-based matching has also been investigated by Rogister et al. [8]. They first perform the rectification of events and then apply the epipolar constraint to event matching. They suggest that events which are in correspondence should have the shortest distance in space and time to the epipolar line. Additionally, they used polarity and ordering constraints to reduce matching ambiguities. The proposed algorithm relies mainly on the temporal agreement of the corresponding event, which could be quite risky. Moreover, single event-to-event matching assumes that object's appearance is the same on the left and right view. This assumption, however, could be hard to obtain dealing with non-rigid and non-frontal motion.

Schraml et al. [7][9] proposed a synchronous approach to event matching. First, events are transformed to an image-like representation and then local window matching is

used as the stereo method. The Normalized Sum of Absolute Differences (NSAD) is used as a cost function to handle the differences in events distribution between left and right view. Matching is performed only on the non-zero pixels, which increases the speed of matching. The proposed algorithm is already used in many DVS applications (e.g. [1][10]). Furthermore, it is included in an embedded software of the product UCOS (smart-eye **U**niversal **CO**unting **S**ensor) [11] for people counting in security applications.

Kogler et al. [4] proposed and compared 3 methods for stereo vision using DVS: area-based matching, event-image-based matching and time-based matching. The first method uses conventional window-based stereo matching applied to the image representation of the events stream. In the second method, the stream is transformed into an image with a 3 state logic (on/off event or no event {– 1 0 1}). The local neighborhood of each event in such an image is used as a matching primitive. The last method uses time for matching cost calculation. Possible matches are weighted according to their distance in time to the reference event. The weights are stored in a dynamic structure called Weighted Matching Image which is updated in time. The weights for events are aggregated in defined time periods.

The field of stereo vision using DVS is not yet extensively investigated. Apart from the work in [10] using synchronous matching, the majority of the proposed methods are rather experimental than fully working algorithms. Furthermore, it is quite hard to compare them as each algorithm was tested not only on different datasets but also using different vision sensors. In general, we can observe that image-based (synchronous) methods usually outscore event-based stereo matching in terms of accuracy and efficiency. Therefore, providing a robust asynchronous stereo matching algorithm still remains a challenge.

3. Adaptive Cooperative Stereo

The goal of this work is to propose a stereo algorithm that is suited for dynamic vision sensors and that would be comparable in terms of accuracy with currently used conventional methods [9][10]. We assume that it is important to preserve the asynchrony of the dynamic vision sensors to fully exploit their potential in efficiency and high-speed. We define three important aspects of the sensor that are considered in the design of our algorithm:

- dynamic matching for spatiotemporal data
- asynchronous processing (data-driven instead of time- driven processing)
- feature matching using data parsimony (edges)

In order to achieve all of the aforementioned points we have to operate directly on the stream of events, thus we aim to propose an event-based method for stereo reconstruction.

In this section we first provide a short description of the original cooperative approach by Marr and Poggio [6] and we explain in which aspects it is suitable for asynchronous matching. Next, we show how the cooperative approach can be adapted for DVS data and we present our algorithm in more details.

3.1. Cooperative approach

Marr and Poggio [6] model the problem of stereo matching with the cooperative network where each node corresponds to an intersection of the left and right sightline. The task of stereo matching is to distinguish the true matches out of all possible matches. In order to do that, Marr and Poggio [6] defined two constraints: uniqueness and smoothness. The former reflects the fact that an object can occupy only one physical position at a particular time. Therefore there is only one true match for a particular feature. The latter says that disparity varies smoothly due to the coherence of the matter. Both constraints are employed in the cooperative network as local neighborhood operations. Nodes at the same disparity level support each other according to the smoothness rule, whereas nodes along the sightline inhibit each other due to the uniqueness constraint. The network iteratively performs those local operations in order to find a global optimum.

There are a few aspects making the cooperative approach very suitable for the DVS data. This method is to a significant extent based on implications from biological stereopsis and therefore adequate for the bio-inspired sensor. For instance, the use of a neural mechanism to achieve a global optimum through multiple local neighborhood operations can be easily employed for dynamic event processing. In addition, the cooperative stereo was designed for matching identical features, thus it can be considered very eligible for matching events.

The applicability of the cooperative approach for the DVS data has been already proven in [5][3].We extend it even further to provide the dynamic adaptation of the network and, therefore, enable asynchronous matching. We suggest storing not only the most recent events but also the history of previous matching results. In order to achieve that, we propose a dynamic cooperative network which is constantly updated as each event contributes new information to it. The result being that this form of the network allows us to perform fully asynchronous matching because each event from the stream can infer its disparity on a basis of information found in the network. Nevertheless, we still use a time-based cost function to assign some initial weights while mapping the possible matches into the network. Additional constraints for false match suppression are already employed in the network by the positive and negative feedback from local neighbors.

Events from the left and right views are mapped to the network, where they receive two types of feedback. Neighbors at the same disparity level implement a cooperative process and give positive feedback to the node, whereas the nodes across the disparity planes compete with each other by negative feedback.
Finally, the node having the highest weight after local neighborhood operations is considered to be the correct match in analogy to winner takes all (WTA) networks).

3.2. The proposed algorithm

The proposed algorithm can be divided into three processing steps: mapping to the network, local network update and winner-takes-all procedure. We assume that events have been previously rectified such that left and right views are geometrically aligned and possible matches can be found on the corresponding epipolar lines. For each event from the input stream, we search for possible matches and assign their initial weights. Next, the network feedback is calculated for each possible match and, finally, a WTA procedure is applied on the nodes to retrieve the correct disparity.

Mapping events to the Cooperative Network

The incoming events are mapped into the cooperative network by their temporal correlation. Therefore, for each event, we search for all possible matches among the most recent events in the opposite view. Additionally, according to the canonical stereo setup, we assume that the true match for any of the left events will always appear on the right view on the left side of the event's position; analogically for the right events.
The matching is done symmetrically for both the left and right events. As the co-occurrence of the events is directly connected with their correspondence, we assume that a higher correlation of events reflects a smaller time difference between them. The matching scores are calculated as in [3] using the following function:

$$f(\Delta t) = \frac{1}{a\Delta t + 1}, \qquad (3)$$

where Δt is the time difference between the reference event and its matching candidate; the parameter a controls the slope of the function.

Cooperative Network – structure, update and disparity calculation
Derived from the cooperative stereo method [6], the possible feature locations in the disparity space are modeled as nodes in the network. In our algorithm the network is a 3D matrix of size (MxNxD), where MxN is the resolution of the sensor and D the considered disparity range. The schematic illustration of the network is shown in Figure 2. The core of the algorithm, however, is in the local neighborhood operations. As previously discussed, we consider two types of neighborhoods: excitatory (positive) and inhibitory (negative). The first one is a square neighborhood of each node within its disparity plane (along x and y directions). The size of the excitatory neighborhood is given as an algorithm parameter.

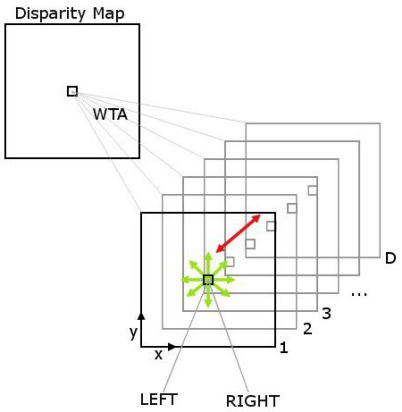

Figure 2: The illustration of the feedback mechanisms used in our cooperative network. Each node of the network is first influenced by an 8-connected spatial neighborhood (green), supporting matches within the same disparity plane. Next, nodes receive the negative feedback from the competing matches across the disparity planes (red).

The positive feedback is a sum of the weights of all neighbors multiplied by the initial weight of the node. The inhibitory neighborhood, however, includes always all the nodes along the sightline, and thus has a constant size of D-1 nodes. Negative feedback applied to the node is done by subtraction of the sum of all inhibitory neighbors multiplied by the given inhibitory factor. Finally, the node values are normalized to have values from 0 to 1. In the last step of the algorithm, we apply a winner-takes-all strategy to the possible matches as the one with the highest value is assumed to be of correct disparity.
The cooperative neural network is constantly changing as the events are generated by the sensor. Each event contributes to the local neighborhood at different disparity levels. Additionally, the node weights decay in time if they were not updated recently. This is done by the global update of the network, performed in given periods of time.

4. Experimental proof of concept

In order to evaluate the algorithm, tests have been first performed using synthetic data depicting different configurations of moving edges. Three sequences were used: (a) edge20 – which consists of one moving edge at a disparity of 20 pixels; (b) 2edges - two edges at different disparities (5 and 20 pixels) moving in two opposite directions; and (c) changDisp with an edge of changing

disparity (from 5 to 20 pixels). The synthetic datasets are shown in Figure 3.

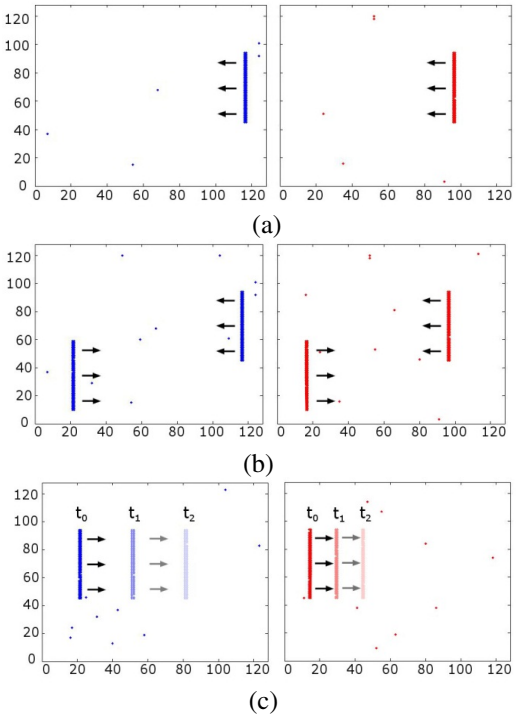

Figure 3: Datasets: (a) *edge20*, (b) *2edges*, *(*c) *changDisp*. The left view is depicted in blue and the right in red. The direction of motion is indicated by arrows. Additionally, in (c) the change of disparity over time is illustrated by showing the position of edges at three different timestamps (t_0, t_1 and t_2).

Table 1 summarizes the tests performed on the synthetic data. It includes details of each sequence such as length and number of events. The results are given by the accuracy and performance. The accuracy is a percentage of events whose disparity agrees with ground-truth (up to one pixel difference is considered as a correct result). The algorithm achieves more than 95% of accuracy. Considering the algorithm's performance, we need to take into account that the processing time is dependent on the events' rate (amount of events in the sequences). Furthermore, the complexity of a scene can influence the processing time due to the higher amount of possible matches to filter out. Therefore, the performance was measured by the events number processed in one time unit (1 second) and this varies around 900 TAEs/s.

According to the results of tests on synthetic data, the algorithm has been positively verified. Therefore, the next step was to test it on real DVS recordings as shown in Figure 6. Two sequences were used depicting a single object in the scene: (a) tool: moving tool at uniform disparity (d = 60) and (b) person: walking person captured by the sensor from overhead mounting.

Table 1: Results of the proposed algorithm on synthetic test data

sequence details			accuracy	Performance
name	length(s)	#events	(%)	TAEs/s
edge20	10,1	51510	98	953,89
changDisp	10,1	51106	97	940,73
2edges	10,1	103020	95	858,50

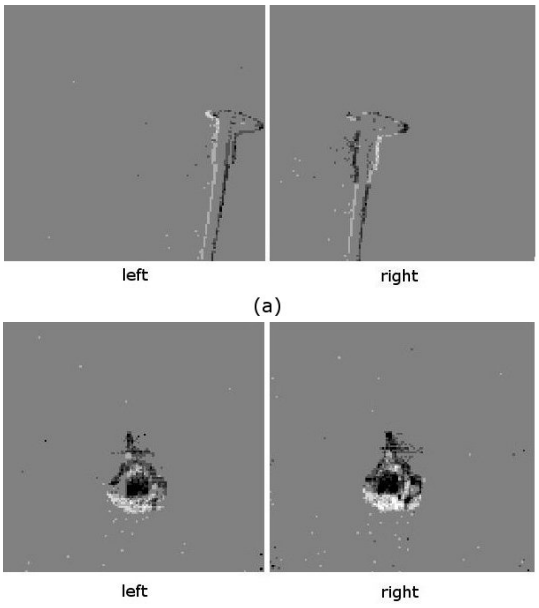

Figure 4: Two simple data sequences from the left and right DVS: (a) moving tool and (b) walking person seen from above.

The obtained disparity maps were compared with results of the algorithm proposed in [10] that uses the conventional NSAD matching on image-like DVS data representation.

As observed in Figure 4(a), some corresponding edges have opposite polarity. If polarity is used as a constraint in the stereo algorithm, the correct (true) matches are eliminated and not considered in matching. That leads to incorrect disparity estimation as visible in Figure 5b. We currently do not consider the polarity constraint. Thus, better results than [10] are obtained, which is illustrated by the disparity histogram with a narrow and high peak at the correct disparity (see Figure 5c). Nevertheless, the polarity may be an additional feature to be considered in future versions of cooperative stereo matching to increase the algorithm's accuracy. The results of stereo matching applied to the person sequence are depicted in Figure 6. In this sequence we have a complex object movement with non-uniform disparity due to the fact that the sensor has been placed in an overhead position. Therefore, we can see that the head is the closest to the sensor, so it has a higher disparity (yellow/red) than the shoulders or legs.

We can also observe that our algorithm is capable of recognizing different disparity levels even in quite challenging object shapes. However, it tends to smooth the transition between disparity levels.

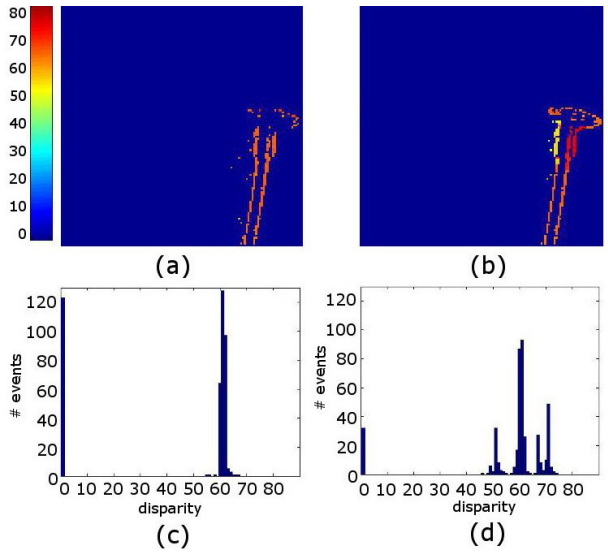

Figure 5: (a) Results of the proposed cooperative algorithm applied to the sequence *tool*. The results are displayed in image-like representation for visual comparison with (b) the algorithm in [10]. The depth (disparity in pixels) is color-coded. The results are also presented by disparity histograms (showing the events number assigned to a particular disparity) for: (c) our cooperative approach and (d) the algorithm in [10].

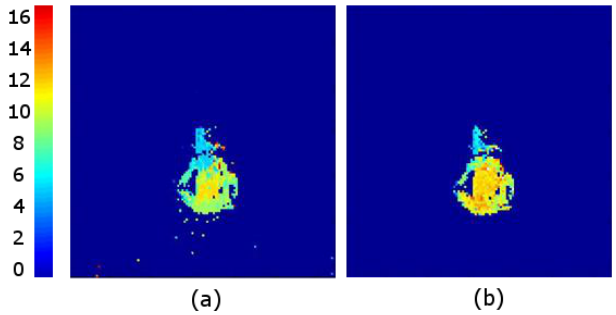

Figure 6: Results of the proposed algorithm (cooperative stereo) applied to the second sequence *person*. Comparison of results displayed in image-like representation of a) our algorithm and b) the algorithm proposed in [10]. The disparity is encoded in color, according to the legend (colorbar).

5. Conclusions

In this work, the problem of asynchronous stereo vision for dynamic vision sensors was addressed. We extend existing methods for event-based processing by using the cooperative approach, which enables spatiotemporal and asynchronous 3D reconstruction.

In the proposed algorithm, stereo matching is modeled with a cooperative network where nodes of the highest activation denote correct disparity. The cooperative aspect is considered as refinement for event-based matching as it not only uses the temporal similarity to match events but also their spatiotemporal neighborhood. The key aspects of our contribution are that the proposed network is dynamic, asynchronous and cooperative. The algorithm was evaluated and tested with a dataset including both real DVS and synthetic data with reference information. These early results proved that the algorithm can successfully perform event-based 3D reconstruction. In the future, we plan to perform a further quantitative and qualitative evaluation of the approach using larger ground truth data.

References

[1] A. Belbachir, A. Nowakowska, S. Schraml, G. Wiesmann, and R. Sablatnig, "Event-driven feature analysis in a 4D spatiotemporal representation for ambient assisted living," in *Proc. ICCV Computer Vision Workshops*, pp. 1570-1577, Barcelona, Spain, 2011.

[2] T. Delbruck, "Frame-free dynamic digital vision," *Proc. Int. Symp. on Secure-Life Electronics, Advanced Electronics and Society*, pp. 21-26, Tokyo, Japan, 2008.

[3] P. Hess, "Low-level Stereo Matching using Event-based Silicon Retina," *semester project report, ETH Zürich*, 2006. Available: http://www.ini.uzh.ch/~tobi/wiki/doku.php?id=publications

[4] J. Kogler, M. Humenberger, and C. Sulzbachner, "Event-Based Stereo Matching Approaches for Frameless Address Event Stereo Data," in *Proc. ISVC*, pp. 674-685, Las Vegas, USA, 2011.

[5] M. Mahowald and T. Delbruck, "Cooperative stereo matching using static and dynamic image features," in *Analog VLSI Implementation of Neural Systems*, C. Mead and M. Ismail, Eds. Boston: Kluwer Academic Publishers, pp. 213-238, 1989.

[6] D. Marr and T. Poggio, "Cooperative Computation of Stereo Disparity," in *Science*, vol. 194, pp. 283-287, 1976.

[7] N. Milosevic, S. Schraml, and P. Schoen,"Smartcam for real-time stereo vision – address-event based Stereo Vision," in Proc. *INSTICC*, pp. 466-471, Barcelona, Spain, 2007.

[8] P. Rogister, R. Benosman, S.H. Ieng, P. Lichtsteiner, and T. Delbruck, "Asynchronous Event-Based Binocular Stereo Matching," *IEEE Trans. Neural Netw. Learn. Syst.,* vol. NNLS-23, pp. 347-353, 2012.

[9] S. Schraml, "Stereo Vision Using Biologically-inspired Vision Sensors," in *Smart Cameras*, A.N. Belbachir, Ed. Springer, pp. 151 -157, 2010.

[10] S. Schraml, A.N. Belbachir, N. Milosevic and P. Schoen, "Dynamic Stereo Vision for Real-time Tracking," in *Proc. of IEEE ISCAS*, pp.1409 – 1412, Paris, France, 2010.

[11] smart eye – UCOS: Universal Counting Sensor, *technology of AIT Austrian Institute of Technology GmbH* http://www.ait.ac.at/uploads/media/Datasheet_UCOS_EN_V8.1_Print_01.pdf